

# CMR API Versioning

We're considering versioning the CMR API and we'd like to get feedback on the preferred approach. We're not sure which approach we want to use yet so if you have feedback we'd appreciate it. The following information is from my winter 2016 ESIP presentation.

## API Versioning Strategies

### URI Path

The version is placed somewhere inside the path of the URL.

`https://cmr.../search/v2/collections`

- Popular (Twitter, etc)
- Problems
  - URIs should be stable. A client has to look inside the URI to know which version of the API it's using. It may have to modify the URL to choose a different version.
  - Granularity problems
    - What does v2 in the URL above refer to? Is it versioning just the search API or searching collections api?

### Query Parameter or Custom Header

This is similar to putting the version in the URI path except it's sent as a query parameter or header

`https://cmr.../search/collections?ver=2`

- Easy to use.
- Problems
  - It's still technically part of the URI and has similar downsides the URI.
  - Granularity issues are worse than the URI. The version is very ambiguous here.

### Content Negotiation

Content Negotiation means that the client specifies versions for the request content type and response content type separately via the built in HTTP mechanisms. HTTP supports the headers Accept and Content-Type to indicate what's being requested or sent.

GET `https://cmr.../search/collections`  
Accept: `application/vnd.nasa.cmr.reference.v2+json`

In this example a collection search is sent and the client has requested the version 2 of the CMR reference response in JSON format. (That doesn't currently exist now. We have a reference format in XML only)

- RESTful
- Request and responses are versioned separately. (Unambiguous granularity)
- URLs stay the same
- Problems
  - More difficult to specify.

Some clients may have difficulty sending HTTP headers. We could offer content negotiation via a URL extension as a workaround for those clients.

GET `https://cmr.../search/collections.ref_v2_json`